Applied Mathematical Sciences, Vol. 7, 2013, no. 13, 603 - 608

Coevolving Solutions of the 3-Satisfiability Problem

Anna Gorbenko

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
gorbenko.ann@gmail.com

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

Coevolutionary genetic algorithms are genetic algorithms that evolve simultaneously two or more populations with coupled fitness. In this paper we consider coevolutionary genetic algorithms for the 3-satisfiability problem.

Keywords: genetic algorithms, coevolution, satisfiability

Genetic algorithms are a class of search algorithms inspired by the biological process of evolution by natural selection. Along with other heuristics (see e.g. [1, 2]), genetic algorithms are extensively used for solution of various hardly formalizable problems (see e.g. [3] – [14]).

Usually, genetic algorithms evolve a population of candidate solutions to a given problem by iteratively applying stochastic search operators. Coevolutionary genetic algorithms are a subclass of genetic algorithms that evolve simultaneously two or more populations with coupled fitness function (see e.g. [15]).

The satisfiability problem is a core problem in contemporary computer science. Satisfiability algorithms widely used for solution of different computationally hard problems (see e.g. [16] – [29]). In particular, genetic algorithms for the 3-satisfiability problem is extensively studied and used for solution of other problems (see e.g. [30] – [38]). In this paper we consider coevolutionary genetic algorithms for the 3-satisfiability problem.

Let $f(z[1], \dots, z[m])$ be a Boolean function. Let

$$W = \{u[1], u[2], \dots, u[n]\}, u[i] \in \{0, 1\}^+, 1 \le i \le n,$$

be a population of chromosomes. We assume that

$$u[i] = u[i, 1] \dots u[i, m], u[i, j] \in \{0, 1\}, 1 \le i \le n, 1 \le j \le m.$$

We can consider $u[i]$ as a solution for $f(z[1], \dots, z[m])$. In particular, we assume that $z[j] = u[i, j]$. We consider only 3CNFs. So, we can assume that

$$f = \wedge_{j=1}^{k} C[j](z[1], \dots, z[m])$$

where $C[j](z[1], \dots, z[m])$ is a clause. We say that $u[i, j]$ is a true assignment for $C[k](z[1], \dots, z[m])$ if $z[j] = u[i, j]$ evaluates $C[k](z[1], \dots, z[m])$ to true. Let $C(u[i, j])$ be the set of clauses such that $C[k](z[1], \dots, z[m]) \in C(u[i, j])$ if and only if $u[i, j]$ is a true assignment for $C[k](z[1], \dots, z[m])$.

During each successive generation, a proportion $\mathcal{P}$ of the existing population is selected to breed a new generation. We assume that $\mathcal{P} = \lceil \frac{n}{2} \rceil$. Individual chromosomes are selected by a fitness function $\mathcal{F}$. After this, the next step is to generate a second generation population of chromosomes. We can use two genetic operators: crossover $\mathcal{C}$ and mutation. In this paper, we consider only crossover. Usually, crossover defines a part of parent chromosome which used for construction of child chromosome. We use standard random crossover. In particular, if $u[i]$ and $u[j]$ are two parent chromosomes, then we obtain two child chromosomes:

$$\mathcal{C}(u[i], u[j]) = u[i, 1] \dots u[i, \mathcal{C}(u[i])]u[j, \mathcal{C}(u[i]) + 1] \dots u[j, m],$$

$$\mathcal{C}(u[j], u[i]) = u[j, 1] \dots u[j, \mathcal{C}(u[j])]u[i, \mathcal{C}(u[j]) + 1] \dots u[i, m].$$

If $u[i[1]], \dots, u[i[p]]$ selected to breed a new generation and $p = 2q$, then

$$\mathcal{C}(u[i[1]], u[i[2]]), \mathcal{C}(u[i[2]], u[i[1]]), \dots,$$

$$\mathcal{C}(u[i[p-1]], u[i[p]]), \mathcal{C}(u[i[p]], u[i[p-1]])$$

is a second generation population of chromosomes. If

$$u[i[1]], \dots, u[i[p]]$$

selected to breed a new generation and $p = 2q + 1$, then

$$u[i[1]], \mathcal{C}(u[i[2]], u[i[3]]), \mathcal{C}(u[i[3]], u[i[2]]), \dots,$$

$$\mathcal{C}(u[i[p-1]], u[i[p]]), \mathcal{C}(u[i[p]], u[i[p-1]])$$

is a second generation population of chromosomes. This generational process is repeated until a termination condition $\mathcal{T}$ has been reached. As $\mathcal{T}$ we consider time function. For simple genetic algorithm (SGA), we can use

$$\mathcal{F}(u[i]) = | \cup_{j=1}^{m} C(u[i,j])|.$$

Now, we consider an example. Let

$$f = (z[1] \vee z[2] \vee z[3]) \wedge (z[1] \vee z[2] \vee \neg z[3]) \wedge (z[1] \vee \neg z[2] \vee z[4]) \wedge$$

$$(z[1] \vee z[2] \vee z[4]) \wedge (z[1] \vee z[3] \vee z[4]) \wedge (z[1] \vee \neg z[2] \vee \neg z[5]) \wedge$$

$$(z[2] \vee z[3] \vee z[4]) \wedge (z[2] \vee \neg z[3] \vee z[4]) \wedge (z[1] \vee z[2] \vee z[5]),$$

$$u[1] = 10000, u[2] = 01000, u[3] = 10100, u[4] = 00010.$$

It is easy to check that

$$\mathcal{F}(u[1]) = 8, \mathcal{F}(u[2]) = 7, \mathcal{F}(u[3]) = 8, \mathcal{F}(u[4]) = 7.$$

In case of SGA, $u[1]$ and $u[3]$ are selected to breed a new generation. Clearly, as a second generation population of chromosomes we can obtain only

$$(10000, 10100, 10000, 10000)$$

or

$$(10000, 10100, 10000, 10100).$$

So, using SGA, f can not be assigned in such a way as to make the formula evaluate to true. However, $u[2]$ and $u[4]$ can give us a true assignment 01010.

To prevent the loss of such solutions we consider a coevolution of solutions. In particular, let

$$\mathcal{F}_k(u[i]) = \max_{\{p[1],\ldots,p[k]\} \subseteq \{1,\ldots,n\}} | \cup_{1 \le j \le m, q \in \{i, p[1], \ldots, p[k]\}} C(u[q,j])|.$$

Using $\mathcal{F}_k$ instead $\mathcal{F}$, we can easily transform SGA into coevolutionary genetic algorithm CGA$[k]$. Selected experimental results are given in Table 1.

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program "Developing the scientific potential of high school" 8.1616.2011.

average number of generations	10^2	10^3	10^4	10^5
average number of true clauses for SGA	57 %	68 %	77 %	81 %
average number of true clauses for CGA[1]	56 %	65 %	84 %	92 %
average number of true clauses for CGA[2]	55 %	63 %	91 %	97 %
average number of true clauses for CGA[3]	54 %	60 %	81 %	88 %

Table 1: Experimental results for different genetic algorithms.

References

[1] A. Gorbenko and V. Popov, Robot Self-Awareness: Occam's Razor for Fluents, *International Journal of Mathematical Analysis*, 6 (2012), 1453-1455.

[2] A. Gorbenko and V. Popov, The Force Law Design of Artificial Physics Optimization for Robot Anticipation of Motion, *Advanced Studies in Theoretical Physics*, 6 (2012), 625-628.

[3] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Exploration of Internal States, *Applied Mathematical Sciences*, 6 (2012), 675-688.

[4] A. Gorbenko, V. Popov, and A. Sheka, Robot Self-Awareness: Temporal Relation Based Data Mining, *Engineering Letters*, 19 (2011), 169-178.

[5] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Finding Regularities, *Applied Mathematical Sciences*, 6 (2012), 6577-6581.

[6] A. Gorbenko and V. Popov, Robot Self-Awareness: Formulation of Hypotheses Based on the Discovered Regularities, *Applied Mathematical Sciences*, 6 (2012), 6583-6585.

[7] A. Gorbenko and V. Popov, Robot Self-Awareness: Usage of Co-training for Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012), 1243-1246.

[8] A. Gorbenko and V. Popov, Robot's Actions and Automatic Generation of Distance Functions for Sequences of Images, *Advanced Studies in Theoretical Physics*, 6 (2012), 1247-1251.

[9] A. Gorbenko and V. Popov, Anticipation in Simple Robot Navigation and Learning of Effects of Robot's Actions and Changes of the Environment, *International Journal of Mathematical Analysis*, 6 (2012), 2747-2751.

[10] A. Gorbenko, A. Lutov, M. Mornev, and V. Popov, Algebras of Stepping Motor Programs, *Applied Mathematical Sciences*, 5 (2011), 1679-1692.

[11] A. Gorbenko and V. Popov, On Face Detection from Compressed Video Streams, *Applied Mathematical Sciences*, 6 (2012), 4763-4766.

[12] A. Gorbenko and V. Popov, Usage of the Laplace Transform as a Basic Algorithm of Railroad Tracks Recognition, *International Journal of Mathematical Analysis*, 6 (2012), 2413-2417.

[13] A. Gorbenko and V. Popov, Face Detection and Visual Landmarks Approach to Monitoring of the Environment, *International Journal of Mathematical Analysis*, 7 (2013), 213-217.

[14] A. Gorbenko and V. Popov, Self-Learning Algorithm for Visual Recognition and Object Categorization for Autonomous Mobile Robots, *Lecture Notes in Electrical Engineering*, 107 (2012), 1289-1295.

[15] W.D. Hillis, Co-evolving parasites improve simulated evolution as an optimization procedure, *Physica D*, 42 (1990), 228-234.

[16] A. Gorbenko and V. Popov, The c-Fragment Longest Arc-Preserving Common Subsequence Problem, *IAENG International Journal of Computer Science*, 39 (2012), 231-238.

[17] A. Gorbenko and V. Popov, On the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 1117-1120.

[18] A. Gorbenko and V. Popov, On the Longest Common Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012), 5781-5787.

[19] A. Gorbenko and V. Popov, Computational Experiments for the Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 5775-5780.

[20] A. Gorbenko and V. Popov, The Binary Paint Shop Problem, *Applied Mathematical Sciences*, 6 (2012), 4733-4735.

[21] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 965-967.

[22] A. Gorbenko, V. Popov, and A. Sheka, Localization on Discrete Grid Graphs, *Lecture Notes in Electrical Engineering*, 107 (2012), 971-978.

[23] A. Gorbenko and V. Popov, The Problem of Selection of a Minimal Set of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 4729-4732.

[24] A. Gorbenko and V. Popov, The Longest Common Parameterized Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012), 2851-2855.

[25] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012), 19-24.

[26] A. Gorbenko and V. Popov, Programming for Modular Reconfigurable Robots, *Programming and Computer Software*, 38 (2012), 13-23.

[27] A. Gorbenko and V. Popov, On the Problem of Placement of Visual Landmarks, *Applied Mathematical Sciences*, 6 (2012), 689-696.

[28] A. Gorbenko, M. Mornev, V. Popov, and A. Sheka, The problem of sensor placement for triangulation-based localisation, *International Journal of Automation and Control*, 5 (2011), 245-253.

[29] A. Gorbenko, M. Mornev, and V. Popov, Planning a Typical Working Day for Indoor Service Robots, *IAENG International Journal of Computer Science*, 38 (2011), 176-182.

[30] A. Gorbenko and V. Popov, SAT Solvers for the Problem of Sensor Placement, *Advanced Studies in Theoretical Physics*, 6 (2012), 1235-1238.

[31] A. Gorbenko and V. Popov, Clustering Algorithm in Mobile Ad Hoc Networks, *Advanced Studies in Theoretical Physics*, 6 (2012), 1239-1242.

[32] A. Gorbenko and V. Popov, The Problem of Finding Two Edge-Disjoint Hamiltonian Cycles, *Applied Mathematical Sciences*, 6 (2012), 6563-6566.

[33] A. Gorbenko and V. Popov, Hamiltonian Alternating Cycles with Fixed Number of Color Appearances, *Applied Mathematical Sciences*, 6 (2012), 6729-6731.

[34] A. Gorbenko and V. Popov, Footstep Planning for Humanoid Robots, *Applied Mathematical Sciences*, 6 (2012), 6567-6571.

[35] A. Gorbenko and V. Popov, Multiple Occurrences Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 6 (2012), 6573-6576.

[36] A. Gorbenko and V. Popov, The Far From Most String Problem, *Applied Mathematical Sciences*, 6 (2012), 6719-6724.

[37] A. Gorbenko and V. Popov, Multi-agent Path Planning, *Applied Mathematical Sciences*, 6 (2012), 6733-6737.

[38] A. Gorbenko and V. Popov, Task-resource Scheduling Problem, *International Journal of Automation and Computing*, 9 (2012), 429-441.

Received: November 1, 2012